3-D Cueing: A Data Filter For Object Recognition

Owen Carmichael and Martial Hebert
{owenc,hebert}@ri.cmu.edu
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh PA 15213

Abstract

This paper presents a novel method for quickly filtering range data points to make object recognition in large 3D data sets feasible. The general approach, called "3D cueing," uses shape signatures from object models as the basis for a fast, probabilistic classification system which rates scene points in terms of their likelihood of belonging to a model. This algorithm, which could be used as a front-end for any traditional 3D matching technique, is demonstrated using several models and cluttered scenes in which the model occupies between 1% and 50% of the data points.

1. Introduction

In this paper, we consider the problem of recognizing three-dimensional models in scenes from range data. Many solutions to the general problem of recognition from 3D free-form surfaces have been proposed and demonstrated successfully [3] [4] [11] [13]. Despite these advancements, however, an issue of scalability remains: the algorithms do not scale well to large data sets with high clutter (see [6] for an analysis of the effect of clutter on recognition algorithms.) More precisely, as the amount of clutter in the scene increases, 3-D matching techniques tend to break down, either because they are unable to represent massive amounts of scene data, or because of fundamental combinatorial limitations inherent to their algorithms. A solution to this problem is to use a fast pre-processing step which would eliminate large portions of the scene from consideration by the matching algorithm. Although such a preprocessing step would reduce the work of the recognition system, it would by no means serve as a substitute.

The notion of pre-processing scene data to facilitate recognition is, of course, well-known in the field of ATR (Automatic Target Recognition) in which fast so-called "cueing" algorithms are used to reduce the large amount of data in an input image to a few regions in which the target has a high probability of being found (e.g., [11]). Although such an approach is common for ATR from intensity images, it has not been applied to the general problem of 3-D object recognition. By analogy to the ATR field, we call the process of quickly filtering a 3D data set for likely points of interest "3-D cueing".

Much of the work in object recognition has focused on the problem of finding corresponding points between one data set (the "scene") and another data set (the "model"). In particular, point-based techniques use compact descriptions of local surface shape around each point in the scene and model (shape "signatures") to find correspondences. Specifically, points with similar signatures are identified as candidate matches.

Following this general approach to matching, finding correspondences is fairly efficient when large portions of the scene and model data sets overlap--an extreme case would be the recognition of an object in a scene containing no clutter. In reality, the sought model typically occupies only a small portion of the scene.

As a result, in scenes in which the object of interest occupies a small percentage of the visible surface area, most of the scene points evaluated for recognition are rejected. In other words, the recognition system may spend most of its time testing points that are clearly dissimilar to the object of interest. For example, consider the task of recognizing the U-shaped pipe fitting in the left scene of Figure 2. It should be obvious, without going through a complex matching procedure, that the points on the adjacent flat surfaces cannot belong to the model, which is curved at all points. Therefore, a 3-D cueing algorithm should filter the points in the scene based on the similarity between their local surface shapes and the local shapes of points on the model surface.

Our approach to cueing is to summarize all the signatures X_i from the model into a classifier f such that $f(X)$ estimates the probability that a signature X belongs to the model. Prior to executing a precise matching algorithm, the classifier is evaluated on an initial selection of points in the scene and only those points to which the classifier assigns

high probability are considered further for recognition (Figure 1).

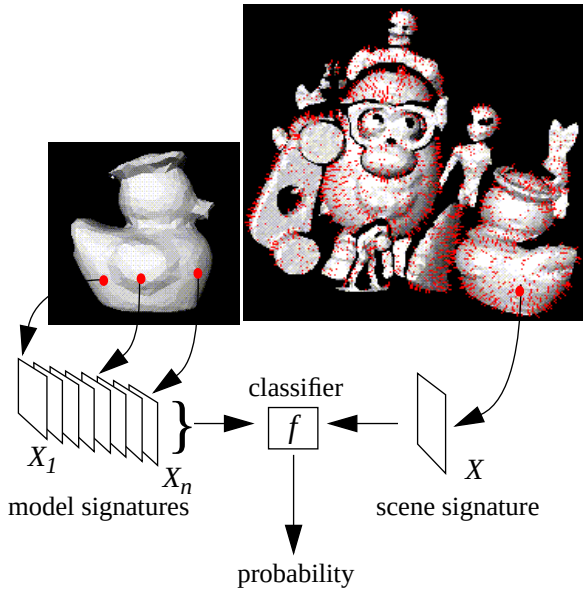

Figure 1. General approach to cueing.

The performance of the classifier is evaluated by computing the ratio of the percentage of points correctly classified as model points to the percentage of the scene that consisted of true model points. Formally, if N_m is the number of model points in the scene according to ground truth, N_c is the number of clutter points in the scene, N'_m is the number of scene points that were correctly classified as model points, and N'_c is the number of scene points incorrectly classified as model points (i.e. points in clutter areas), then we define the performance of the classifier by the ratio $r = \rho'/\rho$, where $\rho' = N'_m/(N'_c + N'_m)$ and $\rho = N_m/(N_c + N_m)$. A high value for r indicates that a large number of true model points are considered for matching, relative to the number of clutter points considered; using cueing, the ratio of true model points to clutter points examined by the recognition system is r times what it would have been if points were picked randomly from the scene. If cueing is performed flawlessly, all the points selected by the filter would lie on the model; therefore $1/\rho$ is the maximum value of r. Quantitatively, our goal is to design a fast point classifier such that:

- The classifier performs better than random selection ($r = 1$) in all cases.
- The classifier often performs close to optimally ($r = 1/\rho$.)

Clearly, the ratio r does not completely summarize the critical effects of cueing; it describes how accurately the filter classifies points as model points, but not how accurately it labels points as clutter. In particular, if the cueing algorithm examines all the points in a particular scene and selects one solitary point as a model point, then $r=1/\rho$ if that point is truly on the model, and the classifier may be said to perform optimally in terms of r; however, if half of the scene consists of model points, the filter clearly does not have a beneficial effect for our recognition system. However, in the results that follow, the points positively labelled by the classifier consist of a reasonable percentage of the target object's points; for this reason we do not concern ourselves with the question of how completely the filter "covers" the model instance.

Obviously, the performance of the classifier degrades in scenes containing multiple instances of very similar surfaces. Nonetheless, it should never perform worse than a random point selection scheme; in other words ρ' should always be greater than ρ. Figure 2 illustrates this point. In the scene on the left, the object of interest (the U-shaped pipe) is very different from the other object in the scene. In this case, the cueing procedure successfully eliminates most of the points from the scene that are outside the desired object. On the other hand, the scene on the right contains more curved objects whose surfaces are similar to that of the model; as a result, the classifier retains more clutter points in the scene. Nevertheless, in both cases, the set of extraneous points to be considered for matching is decreased substantially ($r >> 1$), thus permitting faster and more successful matching.

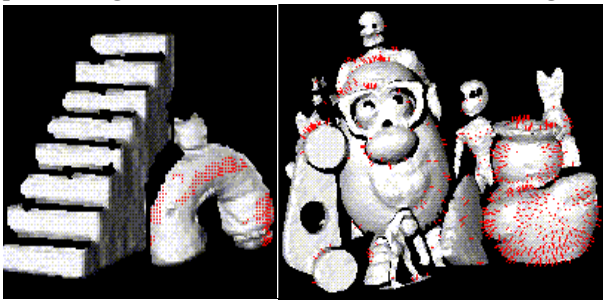

Figure 2. Two examples of 3-D cueing (points retained by the classifier are shown in red.) Left: Only the points on the target object are retained by the cueing classifier. Right: More clutter points are retained because of similar surfaces in the scene.

In the remainder of the paper, we first briefly introduce the shape signature representation used as the basis for our cueing approach. This representation, originally introduced by Johnson [9], is only briefly described here; we refer the reader interested in a complete treatment to [10] and [8]. Based on this representation we then describe the cueing algorithm. Results show our cueing classifier in operation on complex scenes follow.

Although we base the cueing algorithm on a particular surface representation, it is important to note that this procedure is useful whether or not a recognition algorithm based on this representation is used. Techniques based on feature matching would also benefit since most of the potentially misleading features in the background of the scene will be eliminated by the filter.

2. Algorithm

In order to efficiently filter data points based on 3D models, we take a probabilistic classification approach similar to those taken to solve machine learning problems. The classifier we have designed is first trained to recognize model points; then, given a data point from a scene, the classifier predicts whether or not the scene point is part of an instance of the object model by assigning it a probability of model membership. Points given high probability (i.e., above a certain threshold) are assumed to be model points and are returned as the output of the filter.

The descriptor we have chosen to represent local surface shape, the spin image, is a discretized encoding of two of the three cylindrical coordinates of the points on the surface surrounding a particular point. Specifically, for a given point **p**, its spin image X is a 2D array such that the value in bin $X(\alpha,\beta)$ is the number of points on the surface in the neighborhood of **p** that are a distance α away from the surface normal **n** at **p** and a distance β along the normal from **p (**Figure 3**).**

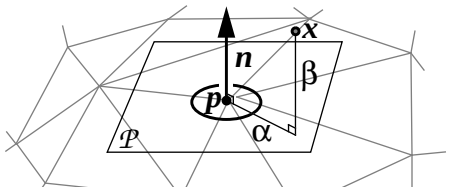

Figure 3. Computation of the two signature coordinates of a point **x** from a basis point **p** (from [9]). Examples of spin images are shown in Figure 6.

Spin images, created by Johnson [8] [9], provide a description of local mesh shape that is both compact and effective in conveying local shape.

The probability model used by the classifier to assign probabilities to scene points is based on the spin images of model points. In this model, the occurrence in the model or scene of a spin image with particular bin values $X(\alpha,\beta)$ is treated as an event; the probability model attempts to describe the distribution of this complex event.

Let X_i be the spin image associated with point i in the scene, and let $i \rightarrow A$ denote the hypothesis that scene point i lies on object A. Then by Bayes' rule,

$$P(i \rightarrow A | X_i) = \frac{P(X_i | i \rightarrow A) P(i \rightarrow A)}{P(X_i)}$$

To find scene points which have the highest probability of belonging to a model, we wish to find the maximum a posteriori (MAP) hypotheses $i \rightarrow A$; that is, the points i which maximize $P(i \rightarrow A | X_i)$. No data is known about the prior probability that any particular scene point i may belong to a model, and no data is known a priori about the probability of a particular spin image X_i occurring in the scene, so uniform distributions $P(i \rightarrow A)$ and $P(X_i)$ are assumed. This reduces the problem to finding the scene points i maximizing $P(X_i | i \rightarrow A)$, i.e. the maximum likelihood hypotheses of $i \rightarrow A$ given X_i over all i's.

Thus, it follows that spin images from the target object model should be analyzed to compute values of $P(X_i | i \rightarrow A)$. It would not be feasible, however, to base cueing calculations on the exact distribution of $P(X_i | i \rightarrow A)$ for model points, simply because this approach does not generalize well to make predictions for data outside of the model. First, since

$$P(X_i | i \rightarrow A) = P(\bigcap_{m,n} X_i(m,n) | i \rightarrow A)$$

and since the set of model spin images form a very sparse set over the sample space of X_i, this joint distribution does not generalize well to accommodate deviations in spin images caused by sensor noise or occlusion. In other words, if most of the bins in a scene spin image take on values that occur in many model spin images, but the other few bins have values which do not occur in the model's spin images, the spin image would have a low probability of being a model point based on the joint distribution above, even though the few deviating bin values could have been caused by self-occlusion, noise, or other factors. Even corresponding points taken from two different scans of the same object will have slightly varying spin images; probabilities based on exact values of the model distribution $P(X_i | i \rightarrow A)$ will be sensitive to these discrepancies. More importantly, computing probabilities of model membership based on exact values of $P(X_i | i \rightarrow A)$ derived from model points consists of comparing the scene spin image to every model spin image in turn to search for an exact replica; the purpose of cueing is precisely to avoid such exhaustive searching.

For these reasons, we assume conditional independence between the pixels in X_i; that is, we assume

$$P(X_i | i \rightarrow A) = \prod_{m,n} P(X_i(m,n) | i \rightarrow A)$$

where $P(X_i(m,n) | i \rightarrow A)$ is the probability that bin $X_i(m,n)$ in a given spin image X_i will equal what it does given that point i is on model A. This assumption of conditional independence of spin image pixels thus reduces our cueing method to naive Bayesian classification [1].

The conditional independence assumption is a critical one because it makes the computation of point probabilities fast; using this probability model we need only compute one probability per spin image bin and take their product. Furthermore, this assumption helps cancel the effects of object occlusion and noise-- if a spin image bin $X_i(m,n)$ contains an unfamiliar number of points, $P(X_i(m,n) | i \rightarrow A)$ will be low for that m and n; how-

ever, if most of the other spin image bins are given high probability, the product $P(X_i|i \to A)$ will still be high on aggregate. Necessity and convenience do not constitute a justification of this assumption, and in particular we have no theoretical reason to believe that spin image bin values would be decorrelated. However, results reported in [1] and [2] illustrate that a probability model which assumes conditional independence models a multivariate sample space surprisingly well considering the reduction of complexity of the probability model.

For a particular pair *(m,n)*, the distribution $P(X_i(m,n)|i \to A)$ is approximated as a discrete histogram. During training, probabilities $P(X_i(m,n)|i \to A)$ are first estimated from the spin images of model points; for every model spin image Y_i, for every bin (m,n), $P(X_i(m,n)|i \to A)$ is incremented by the value of $Y_i(m,n)$, and at the end, each $P(X_i(m,n)|i \to A)$ is divided by the number of model spin images that contributed to it. Then, when a scene point is considered during the classification step, its spin image is constructed, and the above product estimating $P(X_i|i \to A)$ is found by looking up $P(X_i(m,n)|i \to A)$ for every bin (m,n). For numerical reasons, the probability measure computed by the filter for each point *i* is not the actual probability but rather the aggregate likelihood L_i, defined as

$$L_i = -\sum_{m,n} \log(1 - P(X_i(m,n)|i \to A))$$

To process a given scene, a small percentage of its points (usually 10%) are picked at random and probabilities are assigned to them using the classifier. If the probability for a given point is above a certain threshold, the point is labelled as belonging to the model. Furthermore, the points that are nearest neighbors to every predicted model point are considered by the filter; if a neighboring point is labelled as a model point, its neighboring points are analyzed, and so on. Thus, the classification system performs a directed search of the scene, guided by probability assignments. Scene points nearby high-probability points are automatically considered by the classifier so that the geometric contiguousness of model points may be exploited; this leverages our assumption that if a particular point in the scene is on the model, then the points in its immediate neighborhood will probably be on the model as well.

3. Results

The classification system was first tested using a laser triangulation device. The sensor has a maximum range of 3 m and measures a maximum of 160000 points per scan. To conduct early experiments with the triangulation system, we trained the classifier on a VRML model of a PVC pipe U-joint consisting of 600 surface mesh points and placed the object in scenes with various level of clutter. Figure 4 shows the U-joint model and the cueing results obtained in several scenes containing it and two to five other objects. Typical scenes contained between 1000 and 10000 points, 10% of which were randomly chosen as initial points for the filter to process. Points selected by the cueing classifier as members of the U-joint are shown in green in those figures; clearly most of the points are correctly classified, although a few scattered points are selected on the background objects as expected. A series of experiments conducted using this sensor with the U-joint and other models, including the toy duck model shown in Figure 2, is summarized in Figure 5. Here, the accuracy measure $r = \rho'/\rho$ is plotted against ρ, the percentage of the scene covered by the desired object; the curve $r = 1/\rho$ is also charted to show the values of *r* for an optimal classifier. For these scenes, *r* is always substantially greater than it would be if points were selected randomly ($r = 1$) and in many cases performance is close to optimal in terms of *r*. In order to compute ρ' and ρ, all points in input scenes which belonged to the target object were manually labelled as such and points selected by the filter as model points were compared to the true model point set.

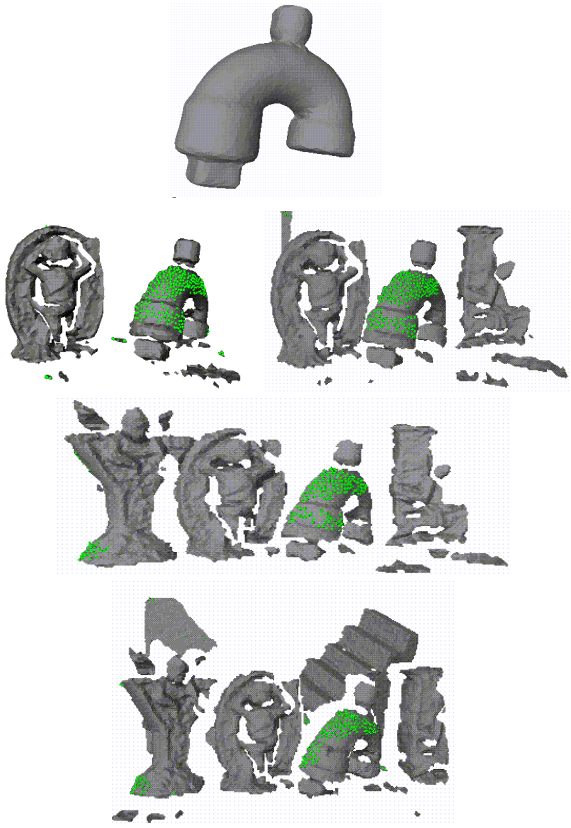

Figure 4. Cueing results on the u-joint model: The u-joint model is shown at the top; the selected points are displayed as green points overlaid on 3-D displays of scenes with 2 to 5 objects.

In order to evaluate the performance of the algorithm in more complex scenes with greater degree of clutter, we used a time-of-flight laser range finder [7] which measures points in a 30-degree by 360-degree field of view with a much longer maximum range (up to 20m in the examples shown in this paper.) We placed the sensor in a cluttered room with known objects and tested the ability of the cueing system to find model points. Two target objects were used in the examples shown here: a plastic deer statue measuring roughly 40 cm by 30

cm by 15 cm (actually a lawn ornament piece!) and a transmission housing from a Ford truck measuring 60 cm by 40 cm by 40 cm (a test object for an application of these recognition techniques to manufacturing problems.) Photos and 3-D models of these objects are shown in Figure 6. The object models contained 1,400 and 6,000 surface points for the transmission housing and for the deer, respectively; spin images for each of these points were used to create the probability models for each object. Experiments involving other objects, such as small statues, were also performed. The purpose of these experiments was to test the cueing under extreme conditions in which the object occupies only about 1% to 5% of the scene.

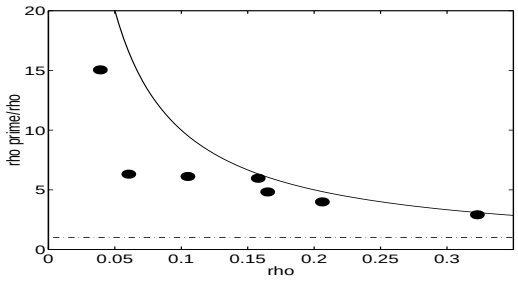

Figure 5. Accuracy results for cueing with triangulation device

Typical data sets from the range finder, containing approximately 2 million data points, are first reduced by retaining only those points that are less than a maximum distance, in these experiments 4 m, from the sensor. This threshold is used only to eliminate large sections of completely planar wall in the background; although walls are always correctly eliminated by the cueing procedure, we wanted to concentrate on areas with highly complex clutter to make the data more difficult to filter. After distance thresholding, the input data set is further reduced to approximately 60,000 points using Garland's decimation algorithm [5]. A random selection of 10% of the reduced data set is then given as input to the cueing program. In typical scenes, this reduced data set covered a physical environment about 8 m by 8m by 2.5 m in volume.

Figure 7 shows a result on a challenging scene containing the deer. Selected points are shown as yellow dots superimposed on the intensity image measured by the range finder. Because the range finder has a 360° field of view, the image is a panoramic view of the environment surrounding the sensor. This intensity image is shown only to illustrate the type of scene used in the experiments; we emphasize that cueing of points is based on the 3-D range data sets. Also shown in Figure 7 is a close-up view of the location of the object, with the selected points displayed in yellow on a texture-mapped 3-D display of the corresponding portion of the scene[1]. For reference, a 3-D view of a large portion of the scene is also included. It is difficult to display the entire scene in 3-D in an intelligible manner; as a result, the 3-D scene view in Figure 7 looks "fragmented" since only those data points whose range is below the distance threshold are displayed. In this example, the object occupies about 1% of the initially selected point set and the final cueing accuracy is $r = 5.24$.

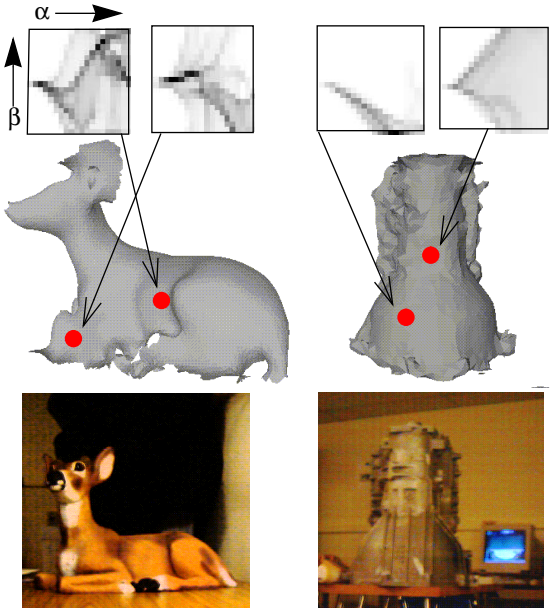

Figure 6. Two objects used for experimentation in scenes with high level of clutter: deer statue (left) and transmission housing (right). Two typical spin images for each object are shown at top; the points giving rise to these spin images are marked in red on the 3D model below, and photos of the objects are shown underneath.

Figure 8 shows a similar example obtained by taking a scan of a large scene containing three copies of the transmission part in different orientations. The figure shows close-ups of the 3-D data of the target objects, as well as an oblique view of the 3-D data. In this example, the three objects together occupy 6.8% of the scene. Experiments conducted on scenes with combinations of the deer and the transmission part show cueing accuracy between 2 and 7 for difficult scenes in which the objects occupy a few percent of the total data set.

A closer examination of the internal operation of the cueing algorithm is shown in Figure 9. Two points, **A** and **B**, are drawn from the scene of Figure 7. **B** belongs to the object, but **A** is in the background clutter. The graphs at the top of Figure 9 show the probabilities $P(X_i(m,n)|i \to A)$ for each bin of the spin images of **A** and **B**, calculated from the deer's probability model. The two horizontal axes in this display are the α and β axes of spin image space; the vertical axis is the probability value for each (α, β) pair. From those probabilities,

1. The very high resolution and large size of these intensity images make it difficult to clearly and accurately display the projections of selected range points due to aliasing effects. For this reason, 3D close-ups of the target objects more accurately depict the distribution of picked points on the model.

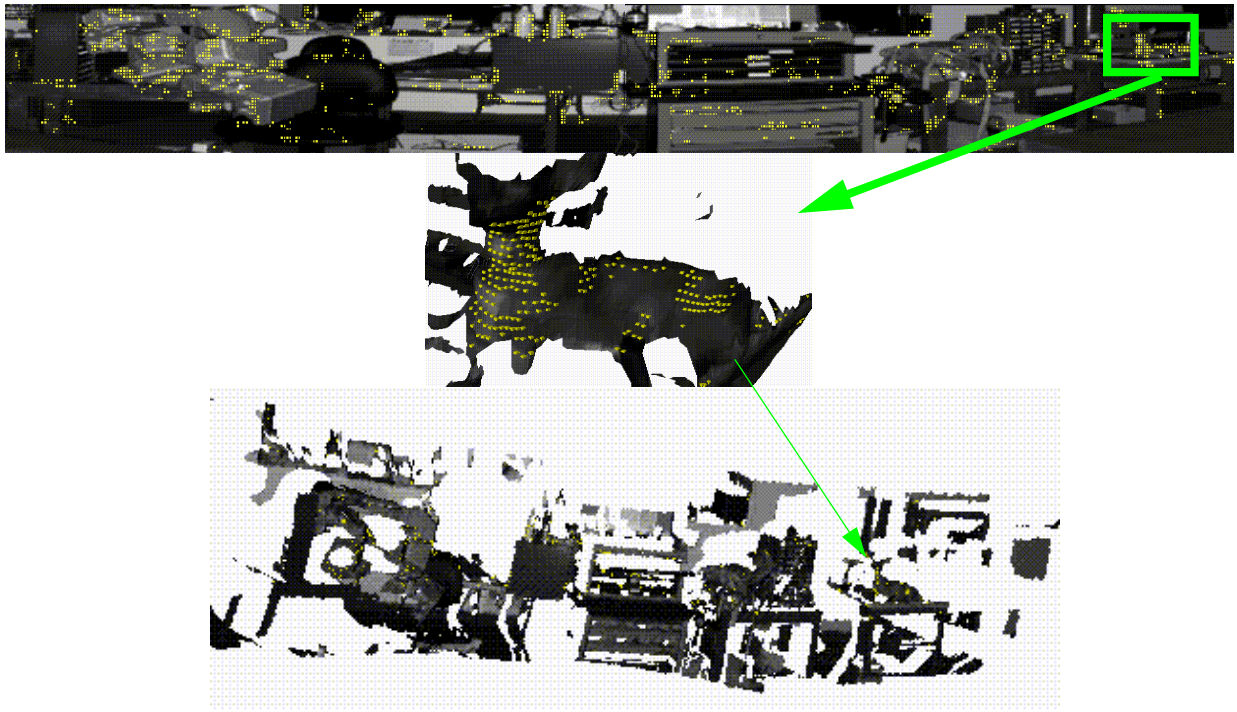

Figure 7. Cueing for the plastic deer in an extreme case of clutter-- the model only occupies about 1% of the scene. Top: Panoramic intensity image from range finder (selected points shown in yellow); Center: Close-up on the object; Bottom: 3-D view of the scene. The 3-D data only is used for cueing; the intensity image is shown for reference only.

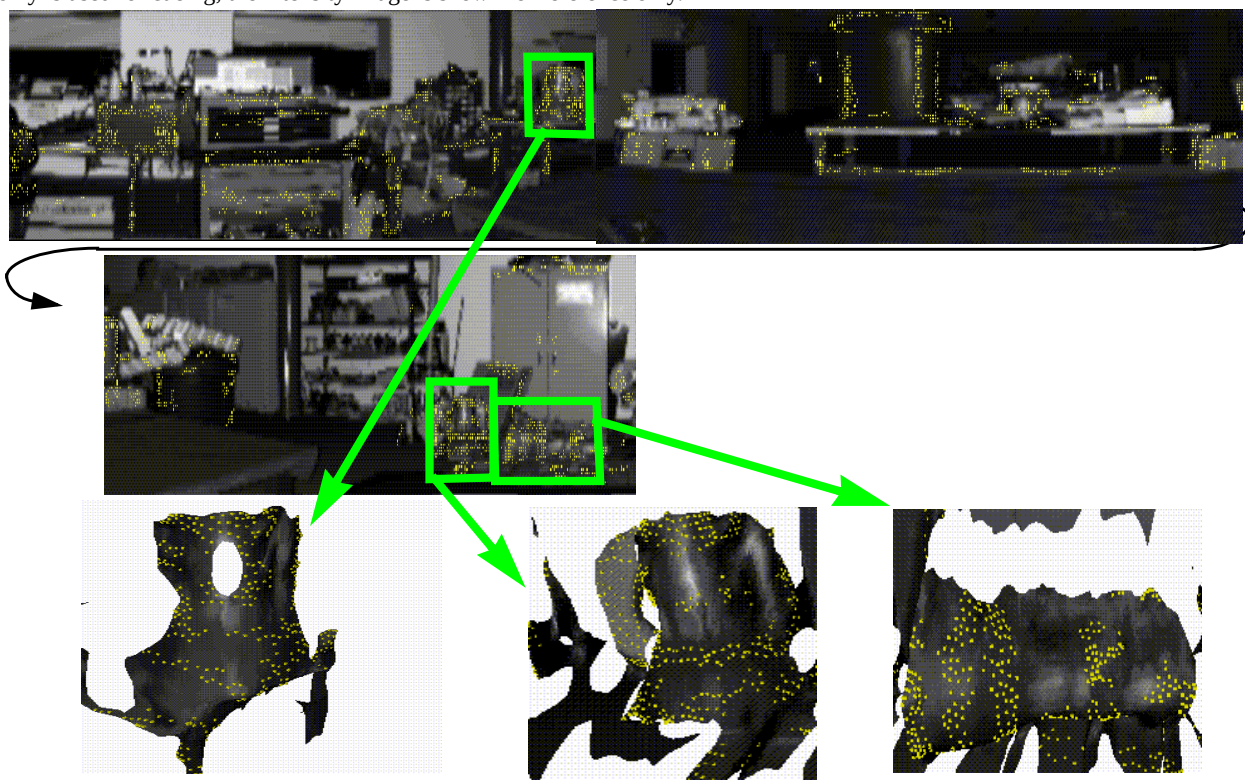

Figure 8. Another difficult cueing scene. The three instances of the target object are highlighted in the panoramic intensity image (shown in two sections for reasons of space) and shown in close-up below. The 3-D data only is used for cueing; the intensity image is shown for reference purposes only.

the aggregate likelihood computed for **A** is 362, as compared to 233 for **B**. Experimentation with these types of data sets was limited due to restricted access to the range finder; additional experiments will be conducted in order to more precisely quantify cueing performance. However, those results are encouraging since they show that in scenes whose high level of clutter precluded the application of traditional matching techniques, the rapid classifier proposed as the basic cueing mechanism can reduce the clutter by a factor of up to seven.

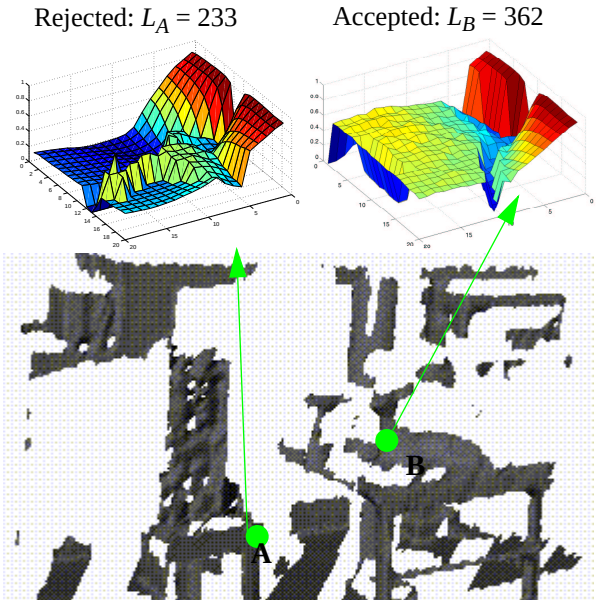

Figure 9. Classification results at two different points in a scene containing the toy deer. The probability maps and aggregate likelihood for the two points A (rejected) and B (accepted) are shown at the top.

Finally, our accuracy measure *r* does not capture another important characteristic of the cueing algorithm. Although many points still remain in the clutter, they are mostly scattered, whereas the points selected on the object form a compact, connected group. As a result, the isolated points in the background will be rapidly eliminated by the matching algorithm because there are not enough matchable points near to them. This remark is based on a qualitative inspection of those results and further work is needed to quantify this observation.

4. Conclusion

Further experiments will be required to completely characterize the performance characteristics of the cueing technique. However, the results shown here prove that the fast data filtering procedure presented here is capable of focusing the points considered by an object recognition algorithm onto an object of interest dramatically in reasonably cluttered environments in which the object of interest covers between 5% and 50% of the scene. Moreover, in exceptionally cluttered scenes the cueing procedure increases point selection accuracy by a factor between 2 and 7. This general approach to range data filtering, which is independent of the specific choice of object shape descriptor and probability model, has the potential to make traditional recognition approaches applicable to large data sets whose sheer volume of data may have caused the techniques to fail otherwise.

References

[1] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[2] P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. Proc. 13th Int'l Conference of Machine Learning (pp. 105-112). 1996.

[3] C.S. Chua and R. Jarvis, "3D free-form surface registration and object recognition", *Int'l J. of Computer Vision*, vol. 17, pp. 77-99, 1996.

[4] C. Dorai, A. Jain. COSMOS - a representation scheme for 3D free-form objects. *IEEE Transaction Pattern on Pattern Analysis and Machine Intelligence*, 19(10): pp. 1115-1130, 1997.

[5] M. Garland, P. S. Heckbert. Surface Simplification Using Quadric Error Metrics. *Proc. SIGGRAPH 97*.

[6] W.E.L. Grimson. *Object recognition by computer: the role of geometric constraints.* MIT Press. 1990.

[7] J. Hancock, D.Langer, M. Hebert, R. Sullivan, D. Ingimarson, E. Hoffman, M. Mettenleitner, C. Froehlich. Active Laser Radar for High Performance Measurements. *Proc. IEEE International Conference on Robotics and Automation*. Leuven, May 1998.

[8] A. Johnson and M. Hebert. Surface matching for object recognition in complex 3-D scenes. to appear in *Image and Vision Computing*. 1998.

[9] A. Johnson and M. Hebert. Efficient multiple model recognition in cluttered 3-D scenes. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*. Santa Barbara, June 1998.

[10] A. Johnson, O. Carmichael, D. Huber, M. Hebert. Toward a general 3-D matching engine: multiple models, complex scenes, and efficient data filtering. *Proc. Image Understanding Workshop*. Monterey, November 1998.

[11] M.W. Koch, M.M. Moya, L.D. Hostetler, R.J. Fogler. Cueing, feature discovery, and one-class learning for automatic target recognition. Neural Networks. Vol. 8, No. 7-8. 1995.

[12] S. Sclaroff and A. Pentland, "Object recognition and categorization using modal matching", *Proc. 2nd CAD-Based Vision Workshop*, pp. 258-265. Champion, Pennsylvania, Feb. 8-11, 1994.

[13] F. Stein and G. Medioni, "Structural indexing: efficient 3-D object recognition", *IEEE Transaction Pattern on Pattern Analysis and Machine Intelligence*, 14(2): pp. 125-145, 1992.